

The Predictability Tax of LEO Handover Predictors

Hannah B. Pasandi
University of California, Berkeley
Berkeley, USA

ABSTRACT

Operators choosing a LEO handover predictor can compare published accuracies but not running costs, because no handover paper reports them. We give cost the reporting discipline accuracy has. The Predictability Tax (PT) is a five-axis cost vector read against the deployment’s budgets, and TAXMETER produces it. Across eight published predictors, two implementations and the live Starlink constellation, accuracy spans 0.29 to 0.38 while per-decision energy spans $289\times$ to $1,460\times$, and the binding budget flips from energy on orbit to latency on the ground. The best predictor depends on where it runs.

1 INTRODUCTION

Starlink reassigns the serving satellite of each terminal on a fifteen-second schedule, at the 12th, 27th, 42nd and 57th second of every minute [17], and the decision of which satellite to hand over to has moved, in three years of publications, from the 3GPP Event A3 comparator with no learned parameters [5] to plan-based MLPs [28], reinforcement learning policies [13] and LSTM and attention stacks with up to 31,041 parameters [6, 11]. The predictor that makes this decision runs on hardware with a power budget. It is an onboard computer in the 2 to 40 W class that Wang *et al.* describe [27], a roaming terminal on a 50 to 150 W vehicle supply [12], or the network side of a direct-to-cell link whose phone pays in battery [8]; every decision costs that platform joules and milliseconds.

The energy and latency of that decision are not reported. The literature evaluates handover predictors by accuracy or F1, lead time and handover latency; in twenty papers since 2024 none reports energy, memory or tail latency per decision (§2). The key tension is that the metric set was built for a comparator whose cost was a constant, and learned predictors turned that constant into a variable spanning three orders of magnitude. Mobile systems papers report an inference architecture together with its energy and latency on the device it targets [25, 31], and work that places foundation models on satellites reports joules per sample [18, 22]; handover prediction has not adopted that norm.

Without energy per decision and tail latency, an operator can choose among predictors only on accuracy, and accuracy does not separate them. The one cost proxy the literature does report is parameter count, and Figure 1 shows it failing in both directions on the workload we measure. A Kalman

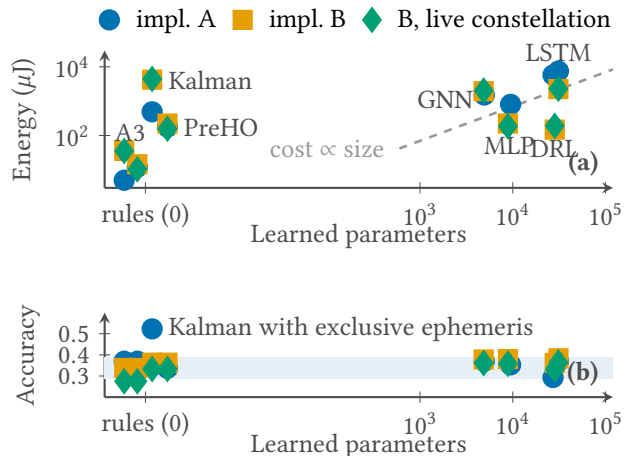


Figure 1: (a) Energy per decision and (b) accuracy against learned parameter count.

filter with no learned parameters costs $4,221 \mu\text{J}$ per decision in one implementation, nearly twice the 31,041-parameter LSTM-attention stack, and a 4,929-parameter graph network costs $13\times$ a 27,137-parameter policy in the other. Accuracy stays between 0.29 and 0.38 from zero parameters to thirty-one thousand on the synthetic workload while energy spreads $289\times$ to $1,460\times$; accuracy cannot rank these predictors and parameter count cannot stand in for their cost.

The handover literature therefore leaves three questions open. (1) What does a published predictor cost per decision on its platform, and does parameter count predict it (§5)? (2) Does the best choice change with the platform (§5)? (3) Which answers survive an independent reimplemention and the live constellation (§5)? To answer them we propose the Predictability Tax, PT, a five-axis cost vector of energy per decision, peak memory, p99.9 latency, dollars per orbit and carbon per gigabyte delivered, each read against the budget of the platform the predictor runs on, whose power ratings and deadlines are published [4, 12, 14].

TAXMETER takes a predictor, a workload and a deployment scenario and emits the five axes plus their worst-case budget fraction. It carries eight predictors re-implemented from their papers and an SGP4 Starlink workload, in two independent implementations, A and B, the second built from the written specification alone; B also ran on the live ephemeris of the

10,679 Starlink satellites in orbit on 3 October 2026, over twenty cities and 28,800 of Starlink’s own decision slots. Two implementations of one specification separate the claims that belong to the predictors from those that belong to the code.

This paper makes four contributions. First, the binding dimension flips from energy on orbit to latency on the ground for every predictor in every run, which makes the same LSTM-attention design the heaviest item on orbit and a comfortable one at a gateway. Second, we price the accuracy trade in both currencies; on live geometry the attention stack buys about 0.1 of accuracy for two decades more energy, and a cascade recovers 81% of that gain for 30% of the energy. Third, the quantitative verdicts do not survive a second implementation, LSTM-attention taking 150% of a 5 mJ satellite budget in A and 45% in B, while the ordering and the flip survive both, measured energy on a third host and batching. Fourth, we release TAXMETER as implementation B with every number behind the figures.¹ We do not propose a predictor, and §6 says which numbers are measured and which are modelled.

Broader impact. Handover prediction is one case of a pattern mobile systems will meet more often, a learned component inside a control loop whose cost lands on someone else’s power budget. We think the discipline proposed here applies wherever that happens, from O-RAN xApps to on-device models that schedule a phone’s radio. For operators, a predictor chosen from accuracy tables may not fit its platform, and the same choice can be right at a gateway and wrong on orbit. For researchers, a cost number means little without the code and host that produced it.

2 THE REPORTING GAP AND RELATED WORK

The twenty LEO handover and pass-window prediction papers we surveyed report accuracy or F1, lead time, handover latency, packet delivery ratio, ping-pong rate and throughput; none reports inference energy, memory, tail latency, dollar cost or carbon. A survey of machine learning in satellite IoT lists two related gaps for the wider field, the oversight of limited computing resources on orbiting satellites and the lack of distinction between space-based and ground-based processing [3]; the satellite and ground budgets of §3 keep those two cases apart. A hybrid rule-based strategy calls learned approaches orders of magnitude costlier at inference without measuring them [2], and PHandover’s authors mitigate the prediction burden rather than measure it [28].

Related work. Starlink measurement characterized its performance and 15 s reconfiguration schedule [15, 17], and the predictors we audit followed [6, 9, 11, 13, 28]. Research on direct-to-device access [19], the wider family

that direct-to-cell belongs to, proposes run-time decisions of the same kind. They range from switching among LEO, GEO and hybrid paths on Doppler, load and application QoS [20] to reinforcement-learning satellite beamforming and ground-based LLM-driven resource management [21], and their costs belong in the same report. Our method follows the inference-energy literature [10, 24] and our budgets the onboard-computing literature [4, 27]. Pantheon and Puffer ran many published schemes behind one interface and found that ranking depends on conditions the originals did not vary [29, 30]; Sadok *et al.* and Mogul and Wilkes argued for reporting disciplines of the same kind [16, 23].

3 THE PREDICTABILITY TAX

For a predictor P on a workload W we define

$$PT(P, W) = (E, M, L, D, C),$$

with E the per-decision energy in joules, M the peak memory in megabytes, L the p99.9 single-decision latency in milliseconds, D the dollar cost per 90-minute orbit and C the operational carbon per gigabyte delivered by the host. A deployment scenario supplies a budget for each axis. The worst-case fraction $PT^* = \max_i (PT_i / \text{budget}_i)$ is the number an operator reads first, and the axis that attains the maximum is the binding dimension. We keep the vector because the axes do not collapse into one; which axis hurts depends on the platform.

Energy. We time one forward pass with Python’s nanosecond counter and multiply the mean wall-clock time by the scenario’s thermal design power, a proxy whose error §6 discusses. We chose p99.9 because a predictor that misses a hard 10 ms xApp deadline once in ten thousand calls is not deployable [14]. Memory is the peak of ten calls under `tracemalloc` plus parameter storage; dollars combine grid energy at 0.12 USD per kWh with memory at cloud prices. Carbon multiplies energy by 350 g CO₂e per kWh for the 2023 U.S. grid [26] or an assumed 50 g in orbit, and never changes a ranking.

Deployment budgets. The satellite scenario assumes a 15 W onboard computer, between the 7.1 W peak measured for orbital edge computing [4] and the 10 to 40 W ratings of radiation-tolerant payload computers [27]. Assuming one decision per terminal every 10 s for 3,000 terminals, 5 mJ per decision holds the predictor to 10% of that computer, with a soft 100 ms deadline and 64 MB of memory. The ground scenario is a Near-RT RIC host at 150 W with the 10 ms end of the xApp control loop that ScalO-RAN measures [14], a 500 mJ energy allowance and 512 MB. PT is a reporting requirement first and a selection rule second, choosing the most accurate predictor that fits every budget.

¹<https://github.com/hanabhp/TaxMeter>, which also logs the eleven places where the written specification left the harness open.

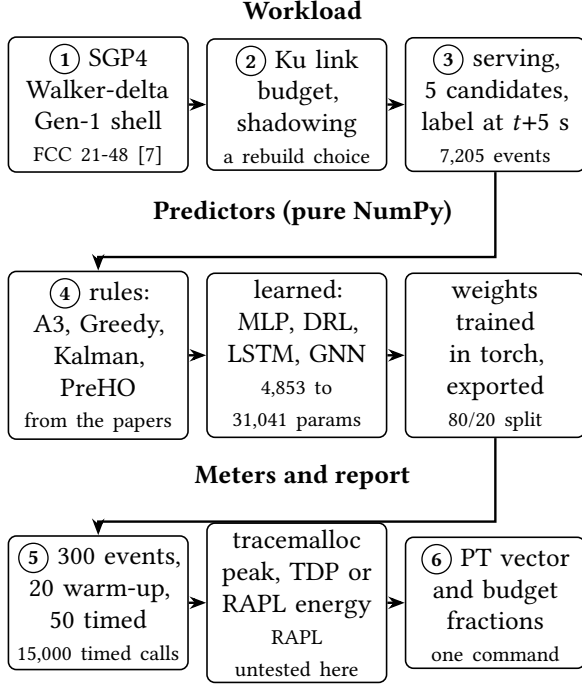


Figure 2: The TAXMETER pipeline from constellation to the five-axis report.

4 TAXMETER AND TWO IMPLEMENTATIONS

TAXMETER is a single-command tool. `taxmeter measure -predictor lstm-attn -scenario satellite -output report.csv` loads the workload, runs the protocol and writes the five axes with their budget fractions. The pipeline has three bands, shown in Figure 2. Stage ① builds the 1,584-satellite 53° shell of the Starlink Gen-1 constellation licensed in FCC order 21-48 [7] and propagates it with SGP4 over four hours for five cities on three continents. Stage ② converts geometry to RSRP through a Ku-band link budget with shadowing and scintillation, and stage ③ emits one event per 10 s with a serving satellite, up to five candidates with RSRP history and ephemeris, and the candidate with the best RSRP five seconds later as label.

Stage ④ holds the eight predictors. Four are rules, the 3GPP A3 comparator [5], a greedy max-RSRP scan, a per-candidate constant-velocity Kalman filter on elevation, and the geometric extrapolation of PreHO [9]. Four are learned, the MLP of PHandover [28], the reinforcement learning policy of DHO [13], an LSTM with two attention heads [6, 11], and a two-layer graph attention network. We re-implemented each in pure NumPy from its paper, the version an engineer who reads the paper obtains first, and trained weights in PyTorch; the two DRL-derived policies are trained supervised, which makes their accuracies lower

bounds and leaves their cost unaffected. A PyTorch forward pass keeps the cost order of the learned predictors with full decision agreement, and caught a first graph layer that scored every candidate alike.

Stages ⑤ and ⑥ run the protocol, 300 sampled events, 20 warm-up and 50 timed calls per event, 15,000 timed inferences in all, with memory read over ten calls and energy as mean wall-clock time times the scenario TDP, then repeat it on a held-out 20%.

Why two implementations. A cost audit has to reimplement the predictors, and every reimplementation makes choices; our parameter counts differ from the published ones for three of the four learned predictors (Table 1). Cost is also far more sensitive to those choices than accuracy, since interpreter overhead rather than arithmetic sets most of it (§5). We therefore built the harness twice.² Implementation A is our original harness; implementation B was built from the written specification alone, without access to A’s code, on a different host.³ Where A and B agree, the claim belongs to the predictors; where they disagree, it belongs to the code, and a third host in §5 shows which.

5 WHAT SURVIVES TWO IMPLEMENTATIONS

Implementations A and B are listed side by side in Table 1; the first three observations below hold in both.

Spread inside the band. In A and in B the accuracies sit between 0.29 and 0.38, apart from A’s Kalman, and the per-decision energies span 5 to 7,499 μJ in A and 15 to 4,221 μJ in B. A3 and Greedy tie on accuracy because the hysteresis gate decides whether to hand over and not where. The synthetic band in Figure 3(a) is flat because a five-second-ahead best target is dominated by geometry every predictor sees and by scintillation none can see; the live run below shows what heavy predictors buy on real geometry.

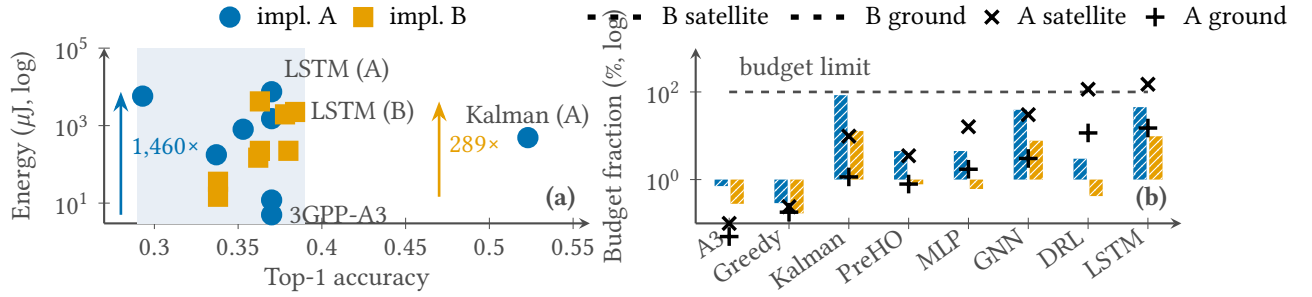
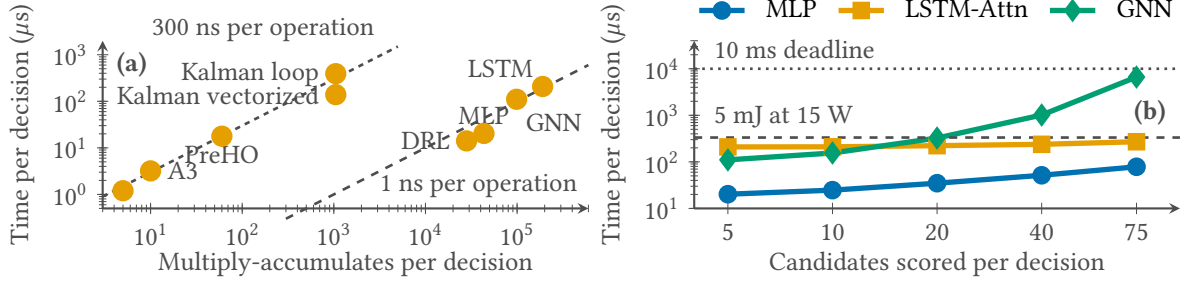
The flip. Reading the costs against the two budgets in Figure 3(b) decides which axis binds; under the satellite scenario energy attains PT_{sat}^* for all eight predictors in A and in B. Under the ground scenario latency attains PT_{gnd}^* for all eight in B and for the five lighter predictors in A, whose three heaviest sit at 11 to 15% of both ground axes. The LSTM-attention design that is the heaviest item on orbit is a comfortable one at a gateway. An accuracy column cannot carry this information, and the flip reproduced in every cell pair we compared.

²Implementation B, the experiments run on it and the figures were produced with the assistance of a generative AI system (Anthropic’s Claude), which also helped draft and edit the text. Every reported number traces to a released run file, and the author takes full responsibility for the content.

³The stated 31,041 for LSTM-attention reproduces exactly as an LSTM(6→64) with a 4→64 candidate embedding, three 64×64 projections and a scalar combine.

Table 1: Eight predictors under two implementations with both deployment budgets.

Predictor	Params		Acc.		E (μ J)		$L_{99.9}$ (ms)		PT_{sat}^* (%)		PT_{gnd}^* (%)		binds
	stated	B	A	B	A	B	A	B	A	B	A	B	
3GPP-A3	0	0	0.370	0.338	5	36	0.005	0.028	0.1	0.7	0.05	0.3	E to L
Greedy	0	0	0.370	0.338	12	15	0.018	0.017	0.2	0.3	0.2	0.2	E to L
Kalman	0	0	0.523	0.363	490	4,221	0.115	1.283	9.8	84.4	1.2	12.8	E to L
PreHO-Geo	0	0	0.337	0.363	177	223	0.079	0.078	3.5	4.5	0.8	0.8	E to L
PHandover-MLP	9,473	8,865	0.353	0.380	807	224	0.172	0.061	16.1	4.5	1.7	0.6	E to L
GNN-DRL	4,929	4,853	0.370	0.378	1,513	1,954	0.293	0.773	30.3	39.1	3.0	7.7	E to E,L
DHO-DRL	27,137	28,326	0.293	0.362	5,775	149	0.824	0.042	115.5	3.0	11.6	0.4	E to E,L
LSTM-Attn	31,041	31,041	0.370	0.384	7,499	2,251	1.106	0.973	150.0	45.0	15.0	9.7	E to E,L

**Figure 3: (a) Accuracy against energy per decision for A and B; (b) worst-case budget fraction under both scenarios.****Figure 4: (a) Time per decision against multiply-accumulate count; (b) time per decision against the number of candidates scored.**

A3 equals Greedy and Kalman’s edge is an input. In A, Kalman reached 0.523 because it alone received elevation and elevation rate. B gives ephemeris to every predictor, and Kalman falls into the band at 0.363 while the learned predictors reach at most 0.384. The assumed premium of heavier predictors over a filter vanished once inputs were equalised, which asks whether the premiums in the original papers are input-access artefacts too.

What did not survive. The energy spread is $1,460\times$ in A and $289\times$ in B, and LSTM-attention takes 150% of the 5 mJ satellite budget in A and 45% in B, crossing it at 7.5 and 2.3 mJ. The rules and the attention stack keep their places, but the

middle of the field reorders; B’s literal Kalman loop costs 281μ s, more than its 150μ s LSTM-attention, and vectorizing the loop cuts it to 99μ s at identical accuracy. Arithmetic does not set this order, as Figure 4(a) shows, the failure nn-Meter found for operation counts on mobile edge devices [31]; time per operation spans 0.5 ns for the dense networks to 370 ns for the Kalman loop. Bootstrap 95% intervals on p99.9 span 6 to 44% of the estimate, and B’s held-out replication moved by up to 36% on a shared vCPU, which we first took for a bug; it was scheduler jitter, and canonical runs now require a pinned core.

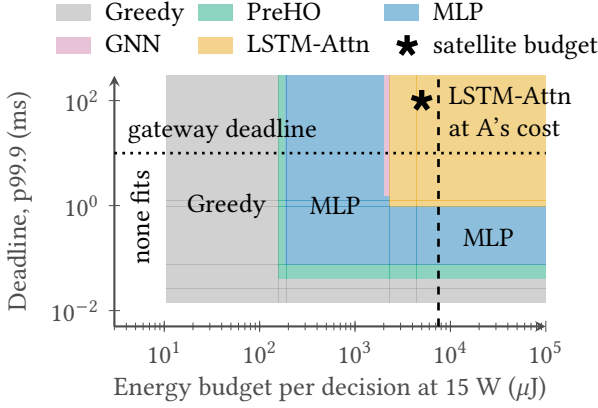


Figure 5: Predictor chosen by the selection rule for each energy budget and deadline.

Live constellation. We re-ran B on the Celestrak ephemeris of the 10,679 Starlink satellites in orbit on 3 October 2026, on Starlink’s 15 s slots for six hours, about four orbits, over twenty cities. The flip holds and LSTM-attention takes 46% of the satellite budget, but accuracy is no longer flat; the comparator scores 0.24 to 0.30 and the attention stack 0.34 to 0.41. Between 14 and 75 satellites are visible above 25°, and scoring all 75, Figure 4(b), multiplies the graph predictor’s cost 60-fold, the MLP’s 3.9-fold and the attention stack’s 1.3-fold.

Measured energy. On an Apple Silicon MacBook Pro we read the chip’s power telemetry with `powermetrics` every 100 ms while each predictor ran single-threaded, subtracting idle power over three interleaved 15 s repetitions. Energy per decision ran from 3.6 μJ for Greedy to 904 μJ for the Kalman loop, 249 $\times$ against 289 $\times$ in B, repetitions agreeing within 10 to 18%. It rank-correlates at 0.98 with the 15 W proxy on the same machine and at 0.93 with column B, while A agrees with either at only 0.57; the same code on two hosts agrees and different code does not, which places the A and B disagreement in the code. The flip holds with measured energy, and LSTM-attention takes 10% of the satellite budget, a third verdict.

Batching. A satellite or gateway serving thousands of terminals can score many decisions per call, removing the per-call overhead of Figure 4(a). With every batched decision matching its one-at-a-time counterpart, batching cuts the attention stack’s time per decision 13-fold at a batch of 32 and 37-fold at 512 but the comparator’s up to 183-fold, and the spread widens from 109 $\times$ to 686 $\times$. At 300 decisions per second a 100 ms deadline allows batches of about 30, where the attention stack takes 6% of the satellite budget, a fourth verdict.

Where each predictor wins. Applying the rule of §3 to every budget and deadline, with the live costs and twenty-city accuracies, gives Figure 5. Below 157 μJ the rule picks the greedy scan, the MLP holds up to 2 mJ, and above 2.3 mJ with a deadline over 1 ms the attention stack wins; tighter deadlines walk the choice back down. The satellite budget lies 2.7 mJ inside the attention stack’s region, and moving its boundary to A’s 7.5 mJ hands that point to the graph predictor.

A cheaper way to buy accuracy. The map suggests a design, not only a choice. On live geometry the geometric rule alone recovers 70% of the attention stack’s accuracy gain over the comparator for 7% of its energy, and a cascade that escalates to the attention stack only when the geometric rule’s two best candidates are within 0.25 dB, 23% of decisions, recovers 81% of the gain for 30% of the energy. Escalating on the current RSRP gap instead buys accuracy only in proportion to energy; the useful signal is where the satellites will be.

6 DISCUSSION

What we do not claim. The energy in Table 1 is wall-clock time times a nameplate TDP, a shortcut the inference-energy literature warns against [10, 24]. On the Mac the proxy overstated every predictor by the same factor of 3.1 to 3.5, which preserves the ordering and the spread but not the joules. We claim the ordering and the flip, which held for every implementation and host we tried. Memory never binds, since no published handover predictor exceeds a few hundred kilobytes.

Implications. Handover papers should report energy per decision on the target platform and p99.9 latency against its deadline beside accuracy, and a cost audit should report its implementation and host with every number. For payload designers, the candidate set, fixed by latitude and constellation density, prices a predictor as much as its architecture does. The study uses public ephemeris and no personal or human-subject data.

What comes next. These results point to a larger program than one audit. First, predictors designed against a budget rather than measured after it, of which the cascade is a first instance and Figure 5 the map of where each design belongs. Second, a shared cost benchmark for LEO predictors in the spirit of Pantheon [30], run on flight-class computers, terminals and phones with external power meters as GreenAuto used [25], and scored against serving satellites identified from dish obstruction maps [1]. Third, the same discipline for the other learned components inside mobile control loops.

7 CONCLUSION

Does a smarter LEO handover predictor pay for itself? We measured eight published predictors against their platforms' budgets. On the synthetic shell accuracy stays within 0.29 to 0.38 while energy spreads $289\times$ to $1,460\times$; on the live constellation two decades of energy buy about 0.1 of accuracy; and in every run the binding budget flips from energy on orbit to latency at a gateway (Figure 3). The multipliers moved between our two implementations, the ordering and the flip did not. Predictor selection must read cost against the deployment, and we suspect reported accuracy premiums shrink once predictors see the same inputs.

REFERENCES

- [1] Ali Ahangarpour, Jinwei Zhao, and Jianping Pan. 2024. Trajectory-based Serving Satellite Identification with User Terminal's Field-of-View. In *Proceedings of the 2nd International Workshop on LEO Networking and Communication*. Association for Computing Machinery, 55–60. <https://doi.org/10.1145/3697253.3697266>
- [2] Khalid Aldubaikhy. 2026. A Low-Complexity Hybrid Handover Strategy for LEO NTN: Balancing Stability and Link Quality. *Sensors* 26, 5 (2026), 1449. <https://doi.org/10.3390/s26051449>
- [3] Alexander Y. Choquenair-Florez, Juan A. Fraire, Hannah B. Pasandi, and Hervé Rivano. 2025. On the role of machine learning in satellite internet of things: A survey of techniques, challenges, and future directions. *Computer Networks* 259 (2025), 111063. <https://doi.org/10.1016/j.comnet.2025.111063>
- [4] Bradley Denby and Brandon Lucia. 2020. Orbital Edge Computing: Nanosatellite Constellations as a New Class of Computer System. In *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*. ACM, 939–954. <https://doi.org/10.1145/3373376.3378473>
- [5] ETSI. 2024. 5G; NR; Radio Resource Control (RRC); Protocol specification (3GPP TS 38.331 version 17.7.0 Release 17). Technical Specification ETSI TS 138 331 V17.7.0. European Telecommunications Standards Institute.
- [6] Dinghe Fan, Shilei Zhou, Jihao Luo, Zijian Yang, and Ming Zeng. 2025. Joint Traffic Prediction and Handover Design for LEO Satellite Networks with LSTM and Attention-Enhanced Rainbow DQN. *Electronics* 14, 15 (2025), 3040. <https://doi.org/10.3390/electronics14153040>
- [7] Federal Communications Commission. 2021. Space Exploration Holdings, LLC Request for Modification of the Authorization for the SpaceX NGSO Satellite System. FCC 21-48, 36 FCC Rod 7995, <https://www.fcc.gov/node/245011>.
- [8] Jorge Garcia-Cabeza, Javier Albert-Smet, Zoraida Frias, Luis Mendo, Santiago Andrés Azcoitia, and Eduardo Yraola. 2025. Direct-to-Cell: A First Look into Starlink's Direct Satellite-to-Device Radio Access Network through Crowdsourced Measurements. *arXiv:2506.00283*. <https://doi.org/10.48550/arXiv.2506.00283>
- [9] Xingqiu He, Zijie Ying, Chaoqun You, and Yue Gao. 2026. PreHO: Predictive Handover for LEO Satellite Networks. *arXiv:2603.07987*. <https://doi.org/10.48550/arXiv.2603.07987>
- [10] Peter Henderson, Jieru Hu, Joshua Romoff, Emma Brunskill, Dan Jurafsky, and Joelle Pineau. 2020. Towards the Systematic Reporting of the Energy and Carbon Footprints of Machine Learning. *Journal of Machine Learning Research* 21, 248 (2020), 1–43.
- [11] Pranathi Kunadi, Navrati Saxena, Yash Bhamre, and Abhishek Roy. 2024. An Efficient Predictive Handover in LEO Satellite Networks using LSTM. In *2024 IEEE International Conference on Communication, Networks and Satellite (COMNETSAT)*. IEEE, 834–839. <https://doi.org/10.1109/comnetsat63286.2024.10862478>
- [12] Dominic Laniewski, Eric Lanfer, Simon Beginn, Jan Dunker, Michael Dückers, and Nils Aschenbruck. 2024. Starlink on the Road: A First Look at Mobile Starlink Performance in Central Europe. In *2024 8th Network Traffic Measurement and Analysis Conference (TMA)*. IEEE, 1–8. <https://doi.org/10.23919/tma62044.2024.10559110>
- [13] Ju-Hyung Lee, Chanyoung Park, Soohyun Park, and Andreas F. Molisch. 2024. Handover Protocol Learning for LEO Satellite Networks: Access Delay and Collision Minimization. *IEEE Transactions on Wireless Communications* 23, 7 (2024), 7624–7637. <https://doi.org/10.1109/twc.2023.3342975>
- [14] Stefano Maxenti, Salvatore D'Oro, Leonardo Bonati, Michele Polese, Antonio Capone, and Tommaso Melodia. 2024. ScalO-RAN: Energy-aware Network Intelligence Scaling in Open RAN. In *IEEE INFOCOM 2024 - IEEE Conference on Computer Communications*. IEEE, 891–900. <https://doi.org/10.1109/infocom52122.2024.10621159>
- [15] François Michel, Martino Trevisan, Danilo Giordano, and Olivier Bonaventure. 2022. A first look at starlink performance. In *Proceedings of the 22nd ACM Internet Measurement Conference*. ACM, 130–136. <https://doi.org/10.1145/3517745.3561416>
- [16] Jeffrey C. Mogul and John Wilkes. 2023. Physical Deployability Matters. In *Proceedings of the 22nd ACM Workshop on Hot Topics in Networks*. ACM, 9–17. <https://doi.org/10.1145/3626111.3628190>
- [17] Nitinder Mohan, Andrew E. Ferguson, Hendrik Cech, Rohan Bose, Prakita Rayyan Renatin, Mahesh K. Marina, and Jörg Ott. 2024. A Multifaceted Look at Starlink Performance. In *Proceedings of the ACM Web Conference 2024*. ACM, 2723–2734. <https://doi.org/10.1145/3589334.3645328>
- [18] Hannaneh B. Pasandi. 2025. WiP: Orbital Edge Intelligence: Real-Time Foundation Model Inference for Earth Observation. In *Proceedings of the 2nd International Workshop on Edge and Mobile Foundation Models*. ACM, 59–62. <https://doi.org/10.1145/3737902.3768362>
- [19] Hannaneh B. Pasandi, Juan A. Fraire, Sylvia Ratnasamy, and Herve Rivano. 2024. A Survey on Direct-to-Device Satellite Communications: Advances, Challenges, and Prospects. In *Proceedings of the 2nd International Workshop on LEO Networking and Communication*. ACM, 7–12. <https://doi.org/10.1145/3697253.3697265>
- [20] Hannaneh B. Pasandi, Mohammad Hosseini, Sina Darabi, Franck Rousseau, and Juan A. Fraire. 2025. Adaptive Orbital Direct-to-Device: Rethinking Satellite IoT Architecture. In *Proceedings of the 20th Workshop on Mobility in the Evolving Internet Architecture*. ACM, 38–43. <https://doi.org/10.1145/3737897.3767293>
- [21] Hannaneh B. Pasandi, Franck Rousseau, Tamer Nadeem, and Sina Darabi. 2025. Beyond Terrestrial: LLM-Orchestrated DtS-IoT Networks for Ubiquitous Edge Intelligence. In *Proceedings of the 2025 ACM Workshop on Access Networks with Artificial Intelligence*. ACM, 41–45. <https://doi.org/10.1145/3737904.3768537>
- [22] Hannaneh B. Pasandi, Zhaowei Tan, Mohammad Hosseini, Sina Darabi, Franck Rousseau, and Juan A. Fraire. 2025. FedSat-LAM: Enabling Large AI Models on Resource-Constrained Satellites via Hierarchical Federated Learning. In *Proceedings of the 2nd International Workshop on Edge and Mobile Foundation Models*. ACM, 19–24. <https://doi.org/10.1145/3737902.3768355>
- [23] Hugo Sadok, Aurojit Panda, and Justine Sherry. 2023. Of Apples and Oranges: Fair Comparisons in Heterogenous Systems Evaluation. In *Proceedings of the 22nd ACM Workshop on Hot Topics in Networks*. ACM, 1–8. <https://doi.org/10.1145/3626111.3628186>
- [24] Arya Tschand, Arun Tejusve Raghunath Rajan, Sachin Idgunji, Anirban Ghosh, Jeremy Holleman, Csaba Kiraly, Pawan Ambalkar, Ritika Borkar, Ramesh Chukka, Trevor Cockrell, Oliver Curtis, Grigori Fursin, Miro Hodak, Hiwot Kassa, Anton Lokhmatov, Dejan Miskovic,

- Yuechao Pan, Manu Prasad Manmathan, Liz Raymond, Tom St. John, Arjun Suresh, Rowan Taubitz, Sean Zhan, Scott Wasson, David Kanter, and Vijay Janapa Reddi. 2025. MLPerf Power: Benchmarking the Energy Efficiency of Machine Learning Systems from μ Watts to MWatts for Sustainable AI. In *2025 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. IEEE, 1201–1216. <https://doi.org/10.1109/hpca61900.2025.00092>
- [25] Xiaolong Tu, Dawei Chen, Kyungtae Han, Onur Altintas, and Haoxin Wang. 2025. GreenAuto: An Automated Platform for Sustainable AI Model Design on Edge Devices. In *Proceedings of the 26th International Workshop on Mobile Computing Systems and Applications*. Association for Computing Machinery, 7–12. <https://doi.org/10.1145/3708468.3711890>
- [26] U.S. Environmental Protection Agency. 2025. eGRID 2023 Summary Tables. Revision 2, <https://www.epa.gov/egrid/summary-data>.
- [27] Haoda Wang, Robert Leo Pendergrast, Kristófer Fannar Björnsson, Anika Somaia, Ezra Landa, Junfeng Yang, and Asaf Cidon. 2025. Above the Clouds: New Software Challenges in Space Computing. In *Proceedings of the 24th ACM Workshop on Hot Topics in Networks*. ACM, 393–401. <https://doi.org/10.1145/3772356.3772396>
- [28] Jiasheng Wu, Shaojie Su, Wenjun Zhu, Xiong Wang, Jingjing Zhang, Xingqiu He, and Yue Gao. 2025. PHandover: Parallel Handover in Mobile Satellite Network. arXiv:2507.07437. <https://doi.org/10.48550/arXiv.2507.07437>
- [29] Francis Y. Yan, Hudson Ayers, Chenzhi Zhu, Sadjad Fouladi, James Hong, Keyi Zhang, Philip Levis, and Keith Winstein. 2020. Learning in situ: a randomized experiment in video streaming. In *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)*. USENIX Association, Santa Clara, CA, 495–511.
- [30] Francis Y. Yan, Jestin Ma, Greg D. Hill, Deepti Raghavan, Riad S. Wahby, Philip Levis, and Keith Winstein. 2018. Pantheon: the training ground for Internet congestion-control research. In *2018 USENIX Annual Technical Conference (USENIX ATC 18)*. USENIX Association, Boston, MA, 731–743.
- [31] Li Lyna Zhang, Shihao Han, Jianyu Wei, Ningxin Zheng, Ting Cao, Yuqing Yang, and Yunxin Liu. 2021. nn-Meter: towards accurate latency prediction of deep-learning model inference on diverse edge devices. In *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services*. Association for Computing Machinery, 81–93. <https://doi.org/10.1145/3458864.3467882>